IN THE SPECIFICATION

Please amend the paragraphs of the specification as follows:

Please replace paragraph 1000 on page 1 with the following amended paragraph:

The present Application for Patent is related to U.S. Patent Application No. 09/957,820, entitled "Method and Apparatus for Coding Bits of Data in Parallel," filed on September 20, 2001, now U.S. Patent No. 6,701,482, issued March 2, 2004 to Salvi et al., assigned to the assignee hereof, and hereby expressly incorporated by reference.

Please replace paragraph 1018 on page 3 with the following amended paragraph:

~~FIGS.~~ FIGs. 5A and 5B are schematic diagrams of a convolutional encoder which implements a specific polynomial generator matrix and for coding eight data bits in parallel according to various embodiments;

Please replace paragraph 1021 on page 3 with the following amended paragraph:

~~FIGS.~~ FIGs. 7B and 7C are diagrams of an interface between an outer convolutional encoder and an interleaver without and with puncturing, respectively, according to various embodiments;

Please replace paragraph 1032 on page 4 with the following amended paragraph:

FIG. 1 is a simplified block diagram of an embodiment of a communications system 100 in which various aspects of the present invention may be implemented. At a transmitter unit 110, traffic data is sent, typically in packets or frames, from a data source 112 to an encoder 114 that formats and codes the data using a particular coding scheme. Encoder 114 typically further performs interleaving (i.e., reordering) of the code bits. A modulator (MOD) 116 then receives, channelizes (i.e., covers), and spreads the coded data to generate symbols that are then converted to one or more analog signals. The analog signals are filtered, (quadrature) modulated, amplified, and upconverted by a ~~transmitter~~ Transmitter (TMTR) 118

to generate a modulated signal, which is then transmitted via an antenna 120 to one or more receiver units.

Please replace paragraph 1033 on page 5 with the following amended paragraph:

At a receiver unit 130, the transmitted signal is received by an antenna 132 and provided to a ~~receiver~~ Receiver (RCVR) 134. Within receiver 134, the received signal is amplified, filtered, downconverted, quadrature demodulated, and digitized to provide samples. The samples are despread, decovered, and demodulated by a ~~demodulator~~ Demodulator (DEMOD) 136 to generate demodulated symbols. A decoder 138 then decodes the demodulated symbols and (possibly) reorders the decoded data to recover the transmitted data. The processing performed by demodulator 136 and decoder 138 is complementary to the processing performed at transmitter unit 110. The recovered data is then provided to a data sink 140.

Please replace paragraph 1036 on page 5 with the following amended paragraph:

The use of CDMA techniques in a multiple access communications system is disclosed in U.S. Patent No. 4,901,307, entitled "SPREAD SPECTRUM MULTIPLE ACCESS COMMUNICATION SYSTEM USING SATELLITE OR TERRESTRIAL REPEATERS," and U.S. Patent No. 5,103,459, entitled "SYSTEM AND METHOD FOR GENERATING WAVEFORMS IN A CDMA CELLULAR TELEPHONE ~~SYSTEM".~~ SYSTEM." Another specific CDMA system is disclosed in U.S. Patent Application Serial No. 08/963,386, entitled "METHOD AND APPARATUS FOR HIGH RATE PACKET DATA TRANSMISSION," filed November 3, 1997, now U.S. Patent No. 6,574,211, issued June 3, 2003 to Padovani et al. (hereinafter referred to as the High Data Rate (HDR) [[HDR]] system). These patents and patent application are assigned to the assignee of the present invention and incorporated herein by reference.

Please replace paragraph 1042 on page 7 with the following amended paragraph:

For clarity, an exemplary embodiment is now described for an encoder used for a downlink data transmission in the communications system described in the aforementioned

U.S. Patent Application Serial No. 08/963,386, now U.S. Patent No. 6,574,211 (i.e., the HDR system). The HDR system employs a concatenated code comprised of an outer convolutional code, interleaving, and an inner convolutional code. The HDR system also defines two packet formats having the properties listed in Table 1.

Please replace paragraph 1046 on page 8 with the following amended paragraph:

FIG. 3 is a diagram of an encoder 300 that implements the outer and inner convolutional codes defined by equations (1) and (2). The data bits u are provided to an outer convolutional encoder 310 that implements equation (1) and generates two outputs y_{oa} and y_{ob}. Within encoder 310, the data bits u are provided to a summer 312 that further couples in cascade with registers [[314a]] 314A through [[314d]] 314D (which are used to implement a set of delays). The outputs from summer 312 and registers 314A, 314B, and 314D are summed by summers 316A, 316B, and 316C to implement the numerator of the second element in the polynomial generator matrix expressed in equation (1). The outputs from registers 314C and 314D are summed by a summer 318 and provided to summer 312 to implement the denominator of the second element in equation (1). The input data bits u are provided as the first output y_{oa} and the output from summer [[316c]] 316C comprises the second output y_{ob}.

Please replace paragraph 1052 on page 10 with the following amended paragraph:

Equations (3) and (4) can be computed for all possible combinations of input data bits and encoder states. For example, for equation (4), the output code bits can be computed for the input vector $U_n = 0...00$ and an encoder state of $X_n = 0...00$, an input vector $U_n = 0...01$ and the encoder state of $X_n = 0...00$, and so on, and an input vector $U_n = 1...11$ and the encoder state of $X_n = 0...00$. The output code bits can then be computed for all possible combination of the input vector U_n and an encoder state of $X_n = 0...01$. The process then continues until all combinations of input vector and encoder state are computed. Equation (3) can also be computed in a similar manner.

Please replace paragraph 1056 on page 11 with the following amended paragraph:

Equations (5) and (6) are used to code one data bit u at a time. A similar set of equations can be derived for coding M data bits in parallel. For example, for coding 8 data bits in parallel (i.e., M = 8), the transpose of the input data vector at time index n can be defined as $U_n^T = [u_{n7} \quad u_{n6} \quad u_{n5} \quad u_{n4} \quad u_{n3} \quad u_{n2} \quad u_{n1} \quad u_{n0}]$, and the transpose of the output code vector can be defined as $Y_n^T = [y_{n7} \quad y_{n6} \quad y_{n5} \quad y_{n4} \quad y_{n3} \quad y_{n2} \quad y_{n1} \quad y_{n0}]$. Using the defined vector notations for U_n and Y_n, equations (5) and (6) can be expressed as:

$$X_{n+1} = FX_n + GU_n \ , \qquad \text{Eq (7)}$$

$$Y_n = HX_n + IU_n \ . \qquad \text{Eq (8)}$$

where F, G, H, and I are vectors and matrices that are dependent on the particular polynomial generator matrix being implemented, the current encoder state X_n, and the input data vector U_n. Equation (7) is used to generate the next encoder state X_{n+1} after M data bits have been coded, and equation (8) is used to generate the encoder outputs Y_n for the input vector U_n.

Please replace paragraph 1061 on page 14 with the following amended paragraph:

Referring back to Table 1, the outer convolutional encoder in the HDR system receives 1018 data bits and four code-tail bits for each packet in packet format 1. If eight bits are coded in parallel, 128 clock cycles are used to code one packet of data. The first 127 clock cycles are used to code 1016 data bits (i.e., 127x8=1016), and the 128th clock cycle is used to code the remaining two data bits and four code-tail bits. The first 127 clock cycles are referred to as the "data ~~phase",~~ phase," and the last clock cycle is referred to as the "code-tail ~~phase".~~ phase."

Please replace paragraph 1074 on page 18 with the following amended paragraph:

FIG. 5B is a schematic diagram of a specific embodiment of a code-tail phase output generator 530 and multiplexers 540A and 540B that implement the code-tail phase of the polynomial generator matrix expressed in equation (1) and for packet formats 1 and 2 shown in

Table 1. Code-tail phase output generator 530 and multiplexers 540A and 540B correspond to code-tail phase output generator 430 and multiplexers 440A and 440B, respectively, in FIG. 4. In this embodiment, code-tail phase output generator 530 is implemented with AND gates 532A through 532J and generates the encoder outputs Y_c and Y_d for the code-tail phase defined in Table 3. Multiplexer [[540a]] 540A is implemented with 2x1 multiplexers 542A through 542F and provides the first encoder output Y_{oa}. Similarly, multiplexer 540B is implemented with 2x1 multiplexers 544A through 544H and provides the second encoder output Y_{ob}.

Please replace paragraph 1098 on page 24 with the following amended paragraph:

For a clearer understanding, a specific design of the interleaver is now described for used with the outer and inner convolutional encoders described above in ~~FIGS.~~ FIGs. 5A, 5B, and 6. In the above encoder designs, the outer convolutional encoder receives and codes 8 data bits in parallel in one clock cycle to generate 16 code bits, and the inner convolutional encoder receives and codes 4 code bits in parallel. In this specific interleaver design, an 8-port memory is employed, with four ports being used for receiving code bits in write operations and four ports being used for providing code bits in read operations. In this design, each port is capable of receiving or providing 8 bits in parallel. Thus, for this specific design, up to 32 code bits can be written to the interleaver in a write operation, and up to 32 code bits can be read from the interleaver in a read operation.

Please replace paragraph 1126 on page 31 with the following amended paragraph:

Some or all of the elements described above for the encoder of the present invention (e.g., multi-bit encoder, input and output interfaces, control unit, encoder state machine, output generator, multiplexer, and so on) can be implemented within one or more ~~application-specific integrated circuits~~ Application Specific Integrated Circuits (ASICs), ~~digital signal processors~~ Digital Signal Processors (DSPs), ~~programmable logic device~~ Programmable Logic Device (PLD), Complex PLD (CPLD), controllers, micro-[[ ]]controllers, microprocessors, other electronic units designed to perform the functions described herein, or a combination thereof.

Some or all of the elements of the encoder of the invention can also be implemented using software or firmware executed on a processor.

Please replace paragraph 1127 on page 31 with the following amended paragraph:

The memories and memory units such as the ones used to implement the interleaver of the present invention can be implemented with various memory technologies such as, for example, ~~random access memory~~ Random Access Memory (RAM), ~~dynamic~~ Dynamic RAM (DRAM), Flash memory, and others. The memory unit can also be implemented with storage elements such as, for example, a hard disk, a CD-ROM drive, and others. Various other implementation of the memory units are possible and within the scope of the present invention.

Please replace paragraph 1130 on page 31 with the following amended paragraph:

FIG. 10 serves as an example of a communications system 1000 that supports a number of users and is capable of implementing at least some aspects and embodiments of the invention. Any of a variety of algorithms and methods may be used to schedule transmissions in system 1000. System 1000 provides communication for a number of cells ~~1020A~~ 1002A through ~~1020G~~ 1002G, each of which is serviced by a corresponding base station ~~1040A~~ 1004A through ~~1040G~~ 1004G, respectively. In the exemplary embodiment, some of base stations ~~1040~~ 1004A to 1004G have multiple receive antennas and others have only one receive antenna. Similarly, some of base stations ~~1040~~ 1004A to 1004G have multiple transmit antennas, and others have single transmit antennas. There are no restrictions on the combinations of transmit antennas and receive antennas. Therefore, it is possible for a base station ~~1040~~ 1004 to have multiple transmit antennas and a single receive antenna, or to have multiple receive antennas and a single transmit antenna, or to have both single or multiple transmit and receive antennas.

Please replace paragraph 1131 on page 32 with the following amended paragraph:

Terminals ~~1060~~ 1006 in the coverage area may be fixed (i.e., stationary) or mobile. As shown in FIG. 1, various terminals ~~1060~~ 1006 are dispersed throughout the system. Each terminal ~~1060~~ 1006 communicates with at least one and possibly more base stations ~~1040~~ 1004

on the downlink and uplink at any given moment depending on, for example, whether soft handoff is employed or whether the terminal is designed and operated to (concurrently or sequentially) receive multiple transmissions from multiple base stations. Soft handoff in CDMA communications systems is well known in the art and is described in detail in U.S. Patent No. 5,101,501, entitled "Method and system for providing a Soft Handoff in a CDMA Cellular Telephone ~~System",~~ System," which is assigned to the assignee of the present invention.

Please replace paragraph 1132 on page 32 with the following amended paragraph:

The downlink refers to transmission from the base station to the terminal, and the uplink refers to transmission from the terminal to the base station. In the exemplary embodiment, some of terminals ~~1060~~ 1006 have multiple receive antennas and others have only one receive antenna. In FIG. 1, base station ~~1040A~~ 1004A transmits data to terminals ~~1060A and 1060J~~ 1006A and 1006J on the downlink, base station ~~1040B~~ 1004B transmits data to terminals ~~1060B and 1060J~~ 1006B and 1006J, base station ~~1040C~~ 1004C transmits data to terminal ~~1060C~~ 1006C, and so on.

Please replace paragraph 1133 on page 32 with the following amended paragraph:

According to an exemplary embodiment, a wireless communication system is adapted for encoding information for transmission using multiple convolutional encoders configured in parallel. Each of the individual encoders has a similar structure and are coupled via an interleaver. The parallel encoders provide a multiple number of outputs, i.e., for two encoders in parallel; the combination provides twice as many output values. A selection is then made at the output for those output values that will be used in further processing. Multiple bits are processed through the parallel encoders. Processing within each encoder is performed in parallel.

Please replace paragraph 1139 on page 34 with the following amended paragraph:

Continuing with FIG. 11, one node of the switch 1501 is coupled to the input. A second node of the switch 1501 is coupled to an exclusive OR (XOR) gate 1504. The output of XOR gate 1504 is coupled to a series of registers or delay elements 1510, 1512, 1514. Each delay element has an associated state, wherein, information stored in delay element 1510 is referred to as in ~~"state 0";~~ "state 0;" information stored in delay element 1512 is referred to as

in ~~"state 1";~~ "state 1;" and information stored in delay element 1514 is referred to as in "state 2." The output of delay element 1510 is identified as ~~"S0";~~ "S0;" the output of delay element 1512 is identified as ~~"S1";~~ "S1;" an the output of delay element 1514 is identified as "S2."

Please replace paragraph 1143 on page 35 with the following amended paragraph:

According to the exemplary embodiment, the turbo encoder has two stages. During the first stage, the frame is read in from an external source. The Cyclic Redundancy Check (CRC) [[CRC]] is also calculated during the first stage. During the second stage the frame is encoded, punctured and repeated. The code rate for the turbo encoder may be 1/3 or 1/5.

Please replace paragraph 1147 on page 35 with the following amended paragraph:

Continuing with FIG. 12, Most Significant Bit (MSB) information from the value of the counter is provided to the ~~add unit~~ ADD circuitry 1302. The ~~add unit~~ ADD circuitry 1302 increments the MSB value of the counter and provides the result to multiply unit 1304. In one embodiment, the resultant value is modified so as to provide only a predetermined number of bits as output. The Least Significant Bit (LSB) information from the value of the counter is provided to Lookup Table (LUT) 1308 and bit reverse unit 1310. The LSB information is used to address the LUT 1308, wherein the value stored in that location is also provided to the multiply unit 1304. The inputs to multiply unit 1304 are multiplied together and the product provided to selection unit 1306. In one embodiment, the multiply unit 1304 provides only a portion of the product as output to selection unit 1306, such as the LSB portion of the product. The bit reverse unit 1310 performs a bit reverse operation, similar to that discussed hereinabove, on the LSB portion of the counter value. The output of the bit reverse unit 1310 is provided to the selection unit 1306. According to the exemplary embodiment, the input to the selection unit 1306 received from the multiply unit 1304 is used as an LSB portion, and the input received from the bit reverse unit 1310 is used as an MSB portion. The selection unit 1306 also determines if the resultant output address is a valid address. If the address is not valid, the selection unit discards the result, wherein on the next counter increment, a new address is generated. Alternate embodiments may implement alternate interleaving schemes applied between the parallel convolutional encoders.

Please replace paragraph 1150 on page 37 with the following amended paragraph:

As discussed hereinabove, the CRC generator 1134 operates during the first state, wherein a 16 bit CRC is computed on the packet currently being processed. A packet includes a payload, a CRC portion and a tail portion. One embodiment supports variable length packets. As the data is read at 16 bits per clock cycle, the CRC generator 1134 computes the CRC every cycle. By the end of the first stage, the CRC is ready. At this point, the CRC is written into the ~~memory storage unit~~ Memory Storage Unit (MEM) 1128 and also into four memory storage devices MEM, 1104 to 1106. Also during the first stage, the information bits are provided to the MEMs 1104 to 1106. The information bits are clocked to the MEMs 1104 to 1106, wherein 16 bits are clocked each clock cycle. Note that in the exemplary embodiment MEMs 1104 to 1106 include four memories, however, alternate embodiments may include alternate numbers of memories. The MEMs 1104 to 1106 receive addressing control information from address generator 1124 and counter 1126, which are each coupled to inputs to a multiplexor 1136. The output of the multiplexer 1136 provides the control signal to the MEMs 1104 to 1106. The address generator 1124 increments the addressing for storage of four values. During a write operation to the MEMs 1104 to 1106, each of the MEMs 1104 to 1106 receives the same address. During a read operation from the MEMs 1104 to 1106, each of the MEMs 1104 to 1106 receives a different address. As illustrated in FIG. 13, the MEM 1128 feeds one of the parallel encoders 1132, while the MEMs 1104 to 1106 feed a second parallel encoder 1144. From the parallel encoders 1132 and 1144, each of which provide output sets of X, Y_0, and Y_1, the output bits are provided to a symbol repetition and puncturing block, such as block 1520 as in FIG. 11.

Please replace paragraph 1156 on page 38 with the following amended paragraph:

The CRC generator 1134 and parallel encoders 1132 and 1144 operate on data at rates greater than 1 bit per clock cycle. The exemplary embodiment implements an AND-XOR tree structure throughout to allow parallel processing. Alternate embodiments may implement any logical structure that recursively implements the Equations (13), (14), and (15). Each AND-XOR tree is given a unique two dimensional array of bits which determine the taps of the AND-XOR tree. For example, consider the parallel encoders 1132, 1144, wherein each includes an internal

3-bit state with different XOR taps for the parity bit outputs, i.e., Y_0, Y_1. Each encoder 1132, 1144 encodes 4 bits per clock cycle in the parallel implementation, wherein a 1/3 rate encoder will produce 12 bits of data per clock, i.e., 4 X bits, 4 Y_0 bits, 4 Y_1 bits. Each output bit is dependent on all 4 input bits as well as the current state. Each encoder includes 3 AND-XOR trees that generate the next two groups of 4 bit output values as well as the next 3 bit state. The X output is directly provided from the input to the encoder, and is not provided through an AND-XOR tree.

Please replace paragraph 1164 on page 41 with the following amended paragraph:

Address generators 1632, 1634, 1636, and 1638 are coupled to MEM 2, MEM 3, MEM 4, and MEM 5, respectively. The MEM 2, MEM 3, MEM 4, and MEM 5 each provide one bit to parallel encoder 1652. The parallel encoder 1652 also provides 4 bit outputs for each of X, Y_0, and Y_1.

Please replace paragraph 1168 on page 42 with the following amended paragraph:

Thus, a novel and improved method and apparatus for encoding multiple bits in parallel, using a recursive method of processing the various outputs has been presented. Addresses are generated for the interleaving operation by use of multiple memory storage devices, wherein a counter is used for generation of interleaver addresses and a mapping is provided to identify invalid addresses. Those of skill in the art would understand that the data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description are advantageously represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof. Those of skill would further appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. The various illustrative components, blocks, modules, circuits, and steps have been described generally in terms of their functionality. Whether the functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans recognize the interchangeability of

hardware and software under these circumstances, and how best to implement the described functionality for each particular application. As examples, the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented or performed with a ~~digital signal processor (DSP)~~ DSP, an ~~application specific integrated circuit (ASIC),~~ ASIC, a ~~field programmable gate array~~ Field Programmable Gate Array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components such as, e.g., registers and ~~FIFO~~ First In First Out (FIFO), a processor executing a set of firmware instructions, any conventional programmable software module and a processor, or any combination thereof designed to perform the functions described herein. The processor may advantageously be a microprocessor, but in the alternative, the processor may be any conventional processor, controller, microcontroller, programmable logic device, array of logic elements, or state machine. The software module could reside in RAM ~~memory~~, flash memory, ~~ROM memory~~ Read Only Memory (ROM), ~~EPROM memory~~ Erasable Programmable ROM (EPROM), ~~EEPROM memory~~ Electrically Erasable Programmable ROM (EEPROM), registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary processor is advantageously coupled to the storage medium so as to read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a telephone or other user terminal. In the alternative, the processor and the storage medium may reside in a telephone or other user terminal. The processor may be implemented as a combination of a DSP and a microprocessor, or as two microprocessors in conjunction with a DSP core, etc.